ACCES I/O PRODUCTS INC
10623 Roselle Street, San Diego, CA 92121
TEL (858)550-9559 FAX (858)550-7322

MODEL PCI-COM485/8

USER MANUAL

FILE: MPCI-COM485-8.E1i

Notice

The information in this document is provided for reference only. ACCES does not assume any liability arising out of the application or use of the information or products described herein. This document may contain or reference information and products protected by copyrights or patents and does not convey any license under the patent rights of ACCES, nor the rights of others.

IBM PC, PC/XT, and PC/AT are registered trademarks of the International Business Machines Corporation.

Printed in USA. Copyright 2001 by ACCES I/O Products Inc, 10623 Roselle Street, San Diego, CA 92121. All rights reserved.

WARNING!!

ALWAYS CONNECT AND DISCONNECT YOUR FIELD CABLING WITH THE COMPUTER POWER OFF. ALWAYS TURN COMPUTER POWER OFF BEFORE INSTALLING A CARD. CONNECTING AND DISCONNECTING CABLES, OR INSTALLING CARDS INTO A SYSTEM WITH THE COMPUTER OR FIELD POWER ON MAY CAUSE DAMAGE TO THE I/O CARD AND WILL VOID ALL WARRANTIES, IMPLIED OR EXPRESSED.

Warranty

Prior to shipment, ACCES equipment is thoroughly inspected and tested to applicable specifications. However, should equipment failure occur, ACCES assures its customers that prompt service and support will be available. All equipment originally manufactured by ACCES which is found to be defective will be repaired or replaced subject to the following considerations.

Terms and Conditions

If a unit is suspected of failure, contact ACCES' Customer Service department. Be prepared to give the unit model number, serial number, and a description of the failure symptom(s). We may suggest some simple tests to confirm the failure. We will assign a Return Material Authorization (RMA) number which must appear on the outer label of the return package. All units/components should be properly packed for handling and returned with freight prepaid to the ACCES designated Service Center, and will be returned to the customer's/user's site freight prepaid and invoiced.

Coverage

First Three Years: Returned unit/part will be repaired and/or replaced at ACCES option with no charge for labor or parts not excluded by warranty. Warranty commences with equipment shipment.

Following Years: Throughout your equipment's lifetime, ACCES stands ready to provide on-site or in-plant service at reasonable rates similar to those of other manufacturers in the industry.

Equipment Not Manufactured by ACCES

Equipment provided but not manufactured by ACCES is warranted and will be repaired according to the terms and conditions of the respective equipment manufacturer's warranty.

General

Under this Warranty, liability of ACCES is limited to replacing, repairing or issuing credit (at ACCES discretion) for any products which are proved to be defective during the warranty period. In no case is ACCES liable for consequential or special damage arriving from use or misuse of our product. The customer is responsible for all charges caused by modifications or additions to ACCES equipment not approved in writing by ACCES or, if in ACCES opinion the equipment has been subjected to abnormal use. "Abnormal use" for purposes of this warranty is defined as any use to which the equipment is exposed other than that use specified or intended as evidenced by purchase or sales representation. Other than the above, no other warranty, expressed or implied, shall apply to any and all such equipment furnished or sold by ACCES.

Table of Contents

Chapter 1: Introduction 1-1
RS485 Balanced Mode Operation 1-1
COM Port Compatibility 1-1
Communication Mode 1-1
Baud Rate Ranges 1-2
Auto-RTS Transceiver Control 1-2
Specifications 1-3
Chapter 2: Installation 2-1
CD Installation 2-1
3.5-Inch Diskette Installation 2-1
Directories Created on the Hard Disk 2-2
Installing the Card 2-4
Windows NT 4.0 Installation Instructions for the PCI-COM485/8 2-5
Chapter 3: Option Selection 3-1
Chapter 4: Address Selection 4-1
Chapter 5: Programming 5-1
Sample Programs 5-1
Windows Programming 5-1
Initialization 5-1
Reception 5-3
Transmission 5-4
Chapter 6: Connector Pin Assignments 6-1
Input/Output Connections 6-1
Appendix A: Application Considerations A-1
Introduction A-1
Balanced Differential Signals A-1
RS485 Data Transmission A-3

List of Figures

Figure 1-1: PCI-COM485/8 Block Diagram . . . . . Page 1-4
Figure 3-1: Simplified Termination Schematic . . . . . Page 3-1
Figure 3-2: PCI-COM485/8 Option Selection Map . . . . . Page 3-3
Figure 3-3: Low-Profile PCI-COM485/8 Option Selection Map . . . . . Page 3-4
Figure A-1: Typical RS485 Two-Wire Multidrop Network . . . . . Page A-3

List of Tables

Table 5-1: Baud Rate Divisor Values . . . . . Page 5-2
Table 6-1: Connector Pin Assignments . . . . . Page 6-1
Table 6-2: Data Cable Wiring . . . . . Page 6-2
Table A-1: Connections Between Two RS422 Devices . . . . . Page A-1
Table A-2: RS422 Specification Summary . . . . . Page A-2

Chapter 1: Introduction

The PCI-COM485/8 Serial Interface Card was designed for effective multipoint transmission in RS485 (EIA485) protocol. The card is 7.83 inches long and may be installed in 5-volt PCI-bus slots of IBM PC or compatible computers. The card features eight independent, asynchronous RS485 serial ports, type 16550 buffered UARTs, and, for Windows compatibility, automatic control to transparently enable/disable the transmission drivers.

RS485 Balanced Mode Operation

The PCI-COM485/8 supports RS485 communications and uses differential balanced drivers for long range and noise immunity. RS485 operation involves switchable transceivers and the ability to support multiple devices on a single "party line". The RS485 specification defines a maximum of 32 devices on a single line. The number of devices served on a single line can be expanded by use of "repeaters".

PCI-COM485/8 also has the capability to add load resistors to terminate the communications lines. RS485 communications requires that one transmitter supply a bias voltage to ensure a known "zero" state when all transmitters are off. Also, receiver inputs at each end of the network should be terminated to eliminate "ringing". The PCI-COM485/8 supports biasing by default and supports termination by jumpers on the card. If your application requires the transmitter to be un-biased, please contact the factory.

COM Port Compatibility

Type 16550 UARTs are used as the Asynchronous Communication Element (ACE). These include a 16-byte transmit/receive buffer to protect against lost data in multitasking operating systems, while maintaining 100 percent compatibility with the original IBM serial port. The system assigns the address(es).

A crystal oscillator is located on the card. This oscillator permits precise selection of baud rate up to 115,200 or, by changing a jumper, up to 921,600 with the standard crystal oscillator.

The driver/receiver used, the SN75176B, is capable of driving extremely long communication lines at high baud rates. It can drive up to ±60 mA on balanced lines and receive inputs as low as 200 mV differential signal superimposed on common mode noise of +12 V or -7 V. In case of communication conflict, the driver/receivers feature thermal shutdown.

Communication Mode

PCI-COM485/8 supports Half-Duplex communications with a 2-wire cable connection. Half-Duplex allows traffic to travel in both directions, but only one way at a time. RS485 communications commonly use the Half-Duplex mode since they share only a single pair of wires.

Baud Rate Ranges

The card has capability for two baud rate ranges and you can select which you wish to use on a port-by-port basis. One range is up to 115,200 baud applications and the other is up to 921,600 baud.

Note

Refer to Table 5-1, Baud Rate Divisor Values on page 5-1 of the manual.

Auto-RTS Transceiver Control

In RS485 communications, the driver must be enabled and disabled as needed, allowing all cards to share a two wire cable. The PCI-COM485/8 card controls the driver automatically. With automatic control, the driver is enabled when data is ready to be transmitted. The driver remains enabled for one additional character's transmission time after data transfer is complete and then is disabled. The receiver is also normally enabled, then disabled during RS485 transmissions, and then re-enabled after transmission is completed (plus one character transmission time). The PCI-COM485/8 automatically adjusts it's timing to the baud rate of the data. (NOTE: Thanks to the automatic control feature, the card is ideal for use in WIN95 applications)

Specifications

Communications Interface

- I/O Connection: 50 Pin SCSI D-Connector
- Serial Ports: Eight cable terminated shielded male D-sub 9-pin connectors with standard IBM AT connectors compatible with RS485 specifications
- Character length: 5, 6, 7, or 8 bits.
- Parity: Even, odd or none.
- Stop Interval: 1, 1.5, or 2 bits.
- Serial Data Rates: Up to 115,200 baud, Asynchronous, A faster range of rates, up to 921,600, is achieved by jumper selection on the card. Type 16550 buffered UART.
- Address: Continuously mappable within 0000 to FFFF (hex) range of PCI bus addresses.
- Receiver Input Sensitivity: ±200 mV, differential input.
- Common Mode Rejection: +12V to -7V
- Transmitter Output Drive Capability: 60 mA, with thermal shutdown.

Environmental

- Operating Temperature Range: 0 °C. to +60 °C.
- Storage temperature Range: -50 °C. to +120 °C.
- Humidity: 5% to 95%, non-condensing.

- Power Required: +5VDC at 125 mA typical, -12VDC at 5 mA typical, +12VDC at 5 mA typical, 750 mW total power consumption.
- Size: 7.8 inches long (198 mm) by 3.8 inches high (97 mm).

Note

The 16550 UART uses a 16-byte first-in-first-out buffer which is programmed through commands sent to the FIFO control register.

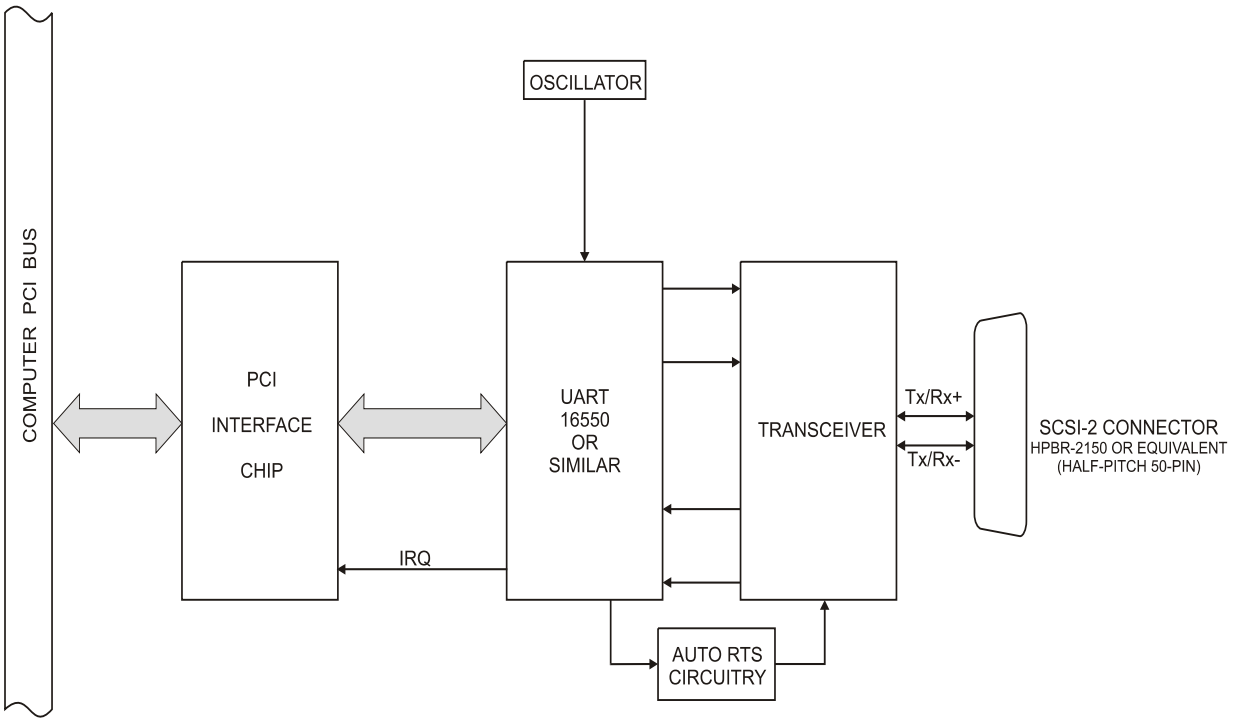

Figure 1-1: PCI-COM485/8 Block Diagram
(Only one serial channel shown)

Chapter 2: Installation

The software provided with this card is contained on either one CD or multiple diskettes and must be installed onto your hard disk prior to use. To do this, perform the following steps as appropriate for your software format and operating system. Substitute the appropriate drive letter for your CD-ROM or disk drive where you see d: or a: respectively in the examples below.

CD Installation

DOS/WIN3.x

1. Place the CD into your CD-ROM drive.
2. Type D : Enter to change the active drive to the CD-ROM drive.
3. Type I N S T A L L Enter to run the install program.
4. Follow the on-screen prompts to install the software for this card.

WIN95/98/NT/2000

1. Place the CD into your CD-ROM drive.
2. The CD should automatically run the install program after 30 seconds. If the install program does not run, click START | RUN and type d:install, click OK or press Enter.
3. Follow the on-screen prompts to install the software for this card.

3.5-Inch Diskette Installation

As with any software package, you should make backup copies for everyday use and store your original master diskettes in a safe location. The easiest way to make a backup copy is to use the DOS DISKCOPY utility.

In a single-drive system, the command is:

D I S K C O P Y A : A : Enter

You will need to swap disks as requested by the system.
In a two-disk system, the command is:

D I S K C O P Y A : B : Enter

This will copy the contents of the master disk in drive A to the backup disk in drive B.

To copy the files on the master diskette to your hard disk, perform the following steps.

1. Place the master diskette into a floppy drive.
2. Change the active drive to the drive that has the diskette installed. For example, if the diskette is in drive A, type A : Enter.
3. Type I N S T A L L Enter and follow the on-screen prompts.

Directories Created on the Hard Disk

The installation process will create several directories on your hard disk. If you accept the installation defaults, the following structure will exist.

[CARDNAME]

Root or base directory containing the SETUP.EXE setup program used to help you configure jumpers and calibrate the card.

DOS\PSAMPLES: A subdirectory of [CARDNAME] that contains Pascal samples.
DOS\CSAMPLES: A subdirectory of [CARDNAME] that contains "C" samples.
Win32\language: Subdirectories containing samples for Win95/98 and NT.

WinRISC.exe

A Windows dumb-terminal type communication program designed for RS422/485 operation. Used primarily with Remote Data Acquisition Pods and our RS422/485 serial communication product line. Can be used to say hello to an installed modem.

ACCES32

This directory contains the Windows 95/98/NT driver used to provide access to the hardware registers when writing 32-bit Windows software. Several samples are provided in a variety of languages to demonstrate how to use this driver. The DLL provides four functions (InPortB, OutPortB, InPort, and OutPort) to access the hardware.

This directory also contains the device driver for Windows NT, ACCESNT.SYS. This device driver provides register-level hardware access in Windows NT. Two methods of using the driver are available, through ACCES32.DLL (recommended) and through the DeviceIOControl handles provided by ACCESNT.SYS (slightly faster).

SAMPLES

Samples for using ACCES32.DLL are provided in this directory. Using this DLL not only makes the hardware programming easier (MUCH easier), but also one source file can be used for both Windows 95/98 and WindowsNT. One executable can run under both operating systems and still have full access to the hardware registers. The DLL is used exactly like any other DLL, so it is compatible with any language capable of using 32-bit DLLs. Consult the manuals provided with your language's compiler for information on using DLLs in your specific environment.

VBACCES

This directory contains sixteen-bit DLL drivers for use with VisualBASIC 3.0 and Windows 3.1 only. These drivers provide four functions, similar to the ACCES32.DLL. However, this DLL is only compatible with 16-bit executables. Migration from 16-bit to 32-bit is simplified because of the similarity between VBACCES and ACCES32.

PCI

This directory contains PCI-bus specific programs and information. If you are not using a PCI card, this directory will not be installed.

SOURCE

A utility program is provided with source code you can use to determine allocated resources at run-time from your own programs in DOS.

PCIFind.exe

A utility for DOS and Windows to determine what base addresses and IRQs are allocated to installed PCI cards. This program runs two versions, depending on the operating system. Windows 95/98/NT displays a GUI interface, and modifies the registry. When run from DOS or Windows3.x, a text interface is used. For information about the format of the registry key, consult the card-specific samples provided with the hardware. In Windows NT, NTioPCI.SYS runs each time the computer is booted, thereby refreshing the registry as PCI hardware is added or removed. In Windows 95/98/NT PCIFind.EXE places itself in the boot-sequence of the OS to refresh the registry on each power-up.

This program also provides some COM configuration when used with PCI COM ports. Specifically, it will configure compatible COM cards for IRQ sharing and multiple port issues.

WIN32IRQ

This directory provides a generic interface for IRQ handling in Windows 95/98/NT. Source code is provided for the driver, greatly simplifying the creation of custom drivers for specific needs. Samples are provided to demonstrate the use of the generic driver. Note that the use of IRQs in near-real-time data acquisition programs requires multi-threaded application programming techniques and must be considered an intermediate to advanced programming topic. Delphi, C++ Builder, and Visual C++ samples are provided.

Findbase.exe

DOS utility to determine an available base address for ISA bus , non-Plug-n-Play cards. Run this program once, before the hardware is installed in the computer, to determine an available address to give the card. Once the address has been determined, run the setup program provided with the hardware to see instructions on setting the address switch and various option selections.

Poly.exe

A generic utility to convert a table of data into an nth order polynomial. Useful for calculating linearization polynomial coefficients for thermocouples and other non-linear sensors.

Risc.bat

A batch file demonstrating the command line parameters of RISCTerm.exe.

RISCTerm.exe

A dumb-terminal type communication program designed for RS422/485 operation. Used primarily with Remote Data Acquisition Pods and our RS422/485 serial communication product line. Can be used to say hello to an installed modem. RISCTerm stands for Really Incredibly Simple Communications TERMinal.

Setup.exe

This program is supplied with the PCI-COM485/8 as a tool for you to use in configuring jumpers and switches on the card. It is menu-driven and provides pictures of the card on the computer monitor. You make simple keystrokes or mouse selections to select functions. The picture on the monitor then changes to show how the jumpers or switches should be placed to effect your choices.

The setup program is a stand-alone program that can be run at any time. It does not require that the card be plugged into the computer. The program is self-explanatory with operation instructions and on-line help.

Installing the Card

The PCI-COM485/8 card can be installed in a five-volt PCI slot of an IBM or compatible computer. Before , carefully read the OPTION SELECTION section of this manual and configure the card according to your requirements. Finally, our SETUP.EXE program will lead you through the process of setting the options on the PCI-COM485/8. The setup program does not set the options. These must be set manually by jumpers on the card.

To Install the Card

1. Turn OFF computer power.
2. Remove the computer cover.
3. Install jumpers from either the Option Selection section of this manual or the suggestions of our SETUP.EXE software program.
4. Install the card in an available PCI-bus slot.
5. Replace the computer cover and turn the computer ON.
6. Enter the CMOS setup program of your system and verify that the PCI plug-and-play option is set appropriately for your system. Systems running Windows95 (or any other PNP-compliant Operating System) should set the CMOS option to OS. Systems running under DOS, WindowsNT 3.51, Windows 3.1, or any other non-PNP-compliant Operating System should set the PNP CMOS option to BIOS or Motherboard. Save the option and continue booting the system.

A "spider" cable is provided to interface between the 50-pin SCSI connector on the card, and your system cabling. Eight individual 9-pin connectors are provided.

Windows NT 4.0 Installation Instructions for the PCI-COM485/8

Unlike Windows95, NT is not a Plug-and-Play compliant Operating System. Windows NT will not auto-detect and install new hardware devices. Therefore, in order to tell Windows NT that you have installed eight new COM ports, you must use the utilities provided by NT in the Control Panel.

Once you have installed the card in the computer and the system is booted to Windows NT 4.0 (Service Pack 3 or higher is recommended), you will also need to install several drivers and utilities provided. These utilities are designed to detect the location of the hardware and report the base address and IRQ assignments for your use. The TOOLS diskette contains these programs.

Place the TOOLS disk in a floppy drive (ex: A:), click the Start Menu, select the Run menu item, and type A:INSTALL [enter]. This will run the installation program from the floppy disk to install the software to your hard-disk.

Once the software installation is complete, run PCIFind.EXE from the root directory created on the hard disk. This program copies NTIOPCI.SYS to the [NT]\SYSTEM32\DRIVERS directory and dynamically loads and runs the driver.

Note

If you previously installed an older version of NTIOPCI.SYS, please delete the file from your [NT]\SYSTEM32\DRIVERS directory prior to running PCIFind.

PCIFind displays the list of COM addresses and the IRQ assigned and adds all pertinent registry entries. The following steps are only necessary if there is a problem running PCIFind and the registry is not updated.

Open the CONTROL PANEL (START|RUN|CONTROL, START|SETTINGS|CONTROL PANEL, or MY COMPUTER|CONTROL PANEL) and execute the PORTS applet.

Click Add...; a dialog box titled "Advanced Settings for New Port" will appear. Select the COM port number of an available COM port, type the Base I/O Port Address of COMA and the IRQ from the PCIFind screen. Click OK.

You have now installed one port in NT. Continue Clicking Add... and selecting addresses for the remainder of the eight ports.

All these steps are performed manually, but could be automated, by making registry entries in various registry keys. Consult your Microsoft provided documentation for more information.

The base addresses and IRQ of the PCI-COM485/8 is provided in the registry under the NTIOPCI key, in a structure of type PCI_COMMON_CONFIG. See the sample program for a demonstration of how to read this structure and extract the addresses in an application program.

Chapter 3: Option Selection

To help you locate the jumpers described in this section, refer to the Option Selection Map at the end of this section. Operation of the serial communications section is determined by jumper installation as described in the following paragraphs.

Terminations

A transmission line should be terminated at the receiving end in its characteristic impedance. Installing a jumper at the locations labeled LDxO applies a 120Ω load across the transmit/receive input/output for RS485 operation.

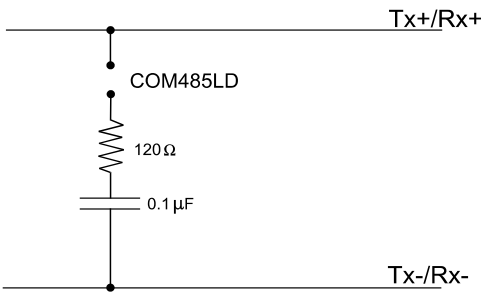

Figure 3-1: Simplified Termination Schematic

In RS485 operations where there are multiple terminals, only the RS485 ports at each end of the network should have terminating impedance as described above. To so terminate the COM A port, place a jumper at the location labeled LDAO. To terminate the COM B, COM C, COM D, COM E, COM F and COM H ports, place jumpers at locations labeled LDBO, LDCO, LDDO, LDEO, LDFO and LDHO respectively.

Also, for RS485 operation, there must be a bias on the TRX+ and TRX- lines. If the PCI-COM485/8 card is not to provide that bias, contact the factory technical support.

Data Cable Wiring

Signal	Pin Connection
Ain/out+	2
Ain/out-	3
100 Ω to Ground	5

Baud Rate Ranges

The jumpers labeled CLK X1 and CLK X4 provide means to select baud rates in either of two ranges. When in the "X1" position, the baud rate range is up to 115,200 baud. When in the CLK X4 position, the baud rate range is 400 to 921,600 baud.

Interrupts

Please note that, in WindowsNT, changes must be made to the system registry to support IRQ sharing. The following is excerpted from "Controlling Multiport Serial I/O Cards" provided by Microsoft in the MSDN library, documentid:mk:@ivt:nt40res/D15/S55FC.HTM, also available in the WindowsNT Resource Kit.

The Microsoft serial driver can be used to control many dumb multiport serial cards. Dumb indicates that the control includes no on-board processor. Each port of a multiport card has a separate subkey under the HKLM\CurrentControlSet\Services\Serial subkey in the registry. In each of these subkeys, you must add values for **DosDevices, Interrupt, InterruptStatus, PortAddress,** and **PortIndex** because these are not detected by the Hardware Recognizer. (For descriptions and ranges for these values, see Regentry.hlp, the Registry help file on the WindowsNT Workstation Resource Kit CD.)

For example, if you have an eight-port PCI-COM485/8 card configured to use address 0xFC00 with an interrupt of 05, the values in the Registry are:

Serial2 Subkey:
PortAddress = REG_DWORD 0xFC00
Interrupt = REG_WORD 5
DosDevices = REG_SZ COM5
InterruptStatus = REG_DWORD 0xFC40
PortIndex = REG_DWORD 1
Indexed = 0

Serial3 Subkey:
PortAddress = REG_DWORD 0xFC08
Interrupt = REG_DWORD 5
DosDevices = REG_SZ COM6
InterruptStatus = REG_DWORD 0xFC40
PortIndex = REG_DWORD 2
Indexed = 0

Serial4 Subkey:
PortAddress =_DWORD 0xFC10
Interrupt = REG_DWORD 5
DosDevices = REG_SZ COM7
InterruptStatus = REG_DWORD 0xFC40
PortIndex - REG_DWORD 3
Indexed = 0

Serial6 Subkey:
PortAddress = REG_DWORD 0xFC20
Interrupt = REG_DWORD 5
DosDevices = REG_SZ COM9
InterruptStatus = REG_DWORD 0xFC40
PortIndex - REG_DWORD 5
Indexed = 0

Serial7 Subkey:
PortAddress = REG_DWORD 0xFC28
Interrupt = REG_DWORD 5
DosDevices = REG_SZ COM10
InterruptStatus = REG_DWORD 0xFC40
PortIndex = REG_DWORD 6
Indexed = 0

Serial8 Subkey:
PortAddress = REG_DWORD 0xFC30
Interrupt = REG_DWORD 5
DosDevices = REG_SZ COM11
InterruptStatus = REG_DWORD 0xFC40
PortIndex = REG_DWORD 7
Indexed = 0

Serial5 Subkey:
PortAddress = REG_DWORD 0xFC18
Interrupt = REG_DWORD 5
Dos Devices = REG_SZ COM8
InterruptStatus = REG_DWORD 0xFC40
PortIndex = REG_DWORD4
Indexed = 0

Serial9 Subkey:
PortAddress = REG_DWORD 0xFC38
Interrupt = REG_DWORD 5
DosDevices = REG_SZ COM12
InterruptStatus = REG_DWORD 0xFC40
PortIndex = REG_DWORD8
Indexed = 0

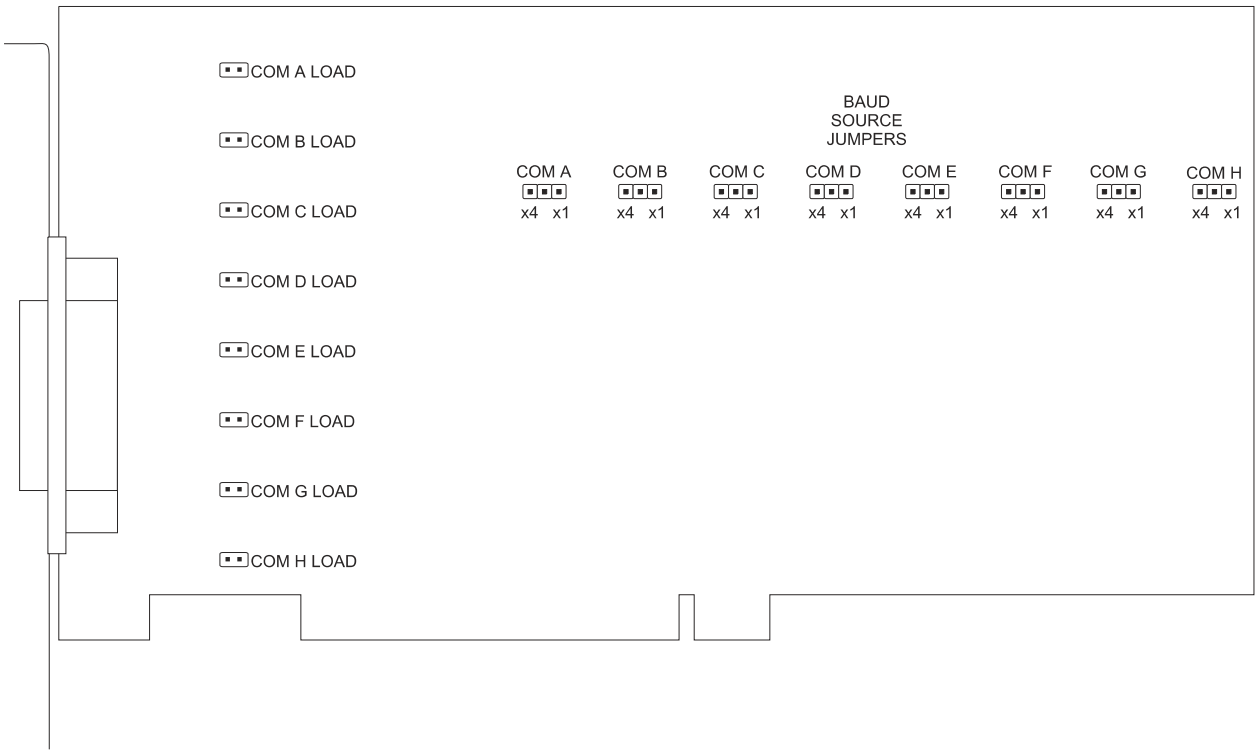

Figure 3-2: PCI-COM485/8 Option Selection Map

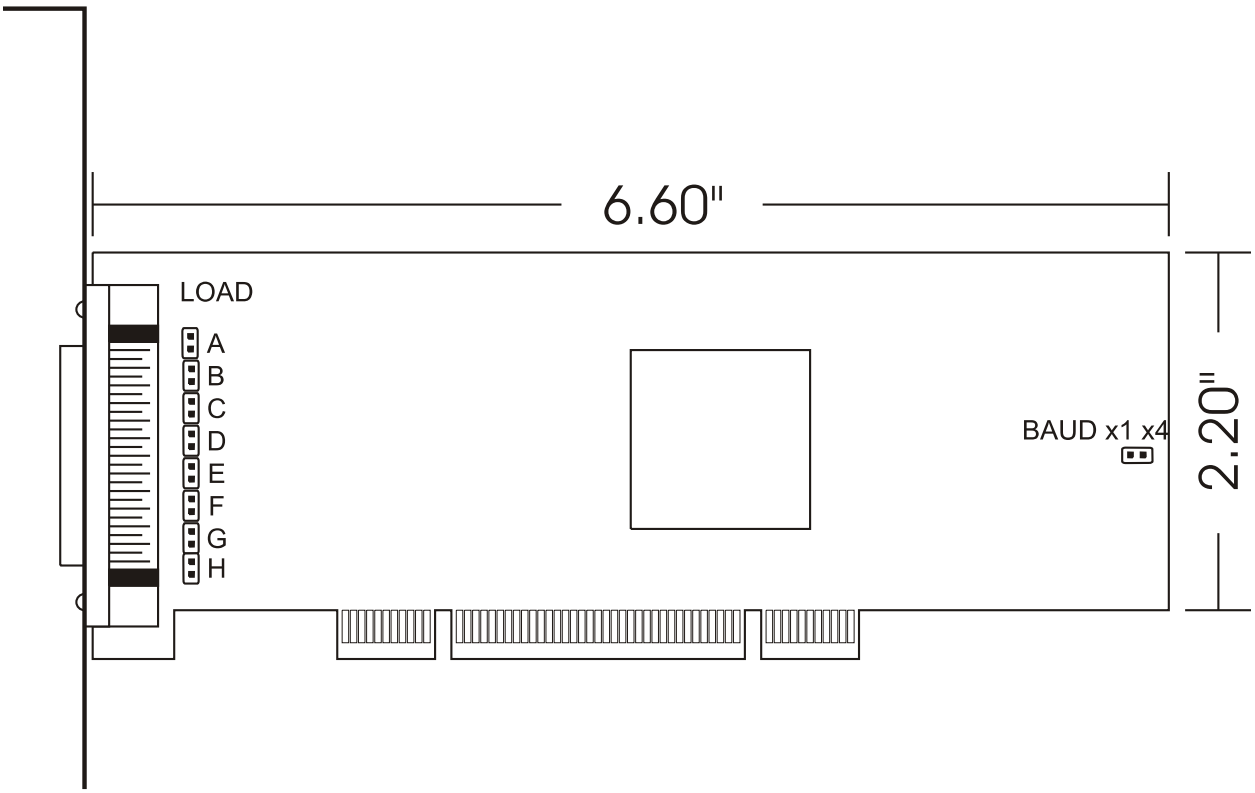

Figure 3-3: Low-Profile PCI-COM485/8 Option Selection Map

Chapter 4: Address Selection

The PCI-COM485/8 card uses one address space. COM A, COM B, COM C, COM D, COM E, COM F, COM G and COM H each occupy eight consecutive register locations. The interrupt register which indicates which port or ports caused the interrupt is located at base address + 64.

PCI architecture is Plug-and-Play. This means that the BIOS or Operating System determines the resources assigned to PCI cards rather than you selecting those resources with switches or jumpers. As a result, you cannot set or change the card's base address. You can only determine what the system has assigned.

To determine the base address that has been assigned, run the PCIFind.EXE, or PCINT utility program provided. This utility will display a list of all of the ACCES cards detected on the PCI bus, the addresses assigned to each function on each of the cards, and the respective IRQs (if any) allotted.

Alternatively, some operating systems (Windows 95/98/2000) can be queried to determine which resources were assigned. In these operating systems, you can use either PCIFind (DOS), PCINT (Windows95/98/NT), or the Device Manager utility from the System Applet of the control panel. The PCI-COM485/8 is installed in the Data Acquisition class of the Device Manager list. Selecting the card, clicking Properties, and then selecting the Resources Tab will display a list of the resources allocated to the card.

The PCI bus supports 64K of I/O space. Your card's addresses may be located anywhere in the 0000 to FFFF hex range.

PCIFind uses the Vendor ID and Device ID to search for your card, then reads the base address and IRQ.

If you want to determine the base address and IRQ yourself, use the following information.

The Vendor ID for this card is 494F. (ASCII for "IO")
The Device ID for the PCI-COM485/8 is 1069h.

Chapter 5: Programming

Sample Programs

There are sample programs provided with the PCI-COM485/8 card in C, Pascal, QuickBASIC, and several Windows languages. DOS samples are located in the DOS directory and Windows samples are located in the WIN32 directory.

Windows Programming

The PCI-COM485/8 card installs into Windows as COM ports. Thus the Windows standard API functions can be used. In particular:

- CreateFile() and CloseHandle() for opening and closing a port.
- SetupComm(), SetCommTimeouts(), GetCommState(), and SetCommState() to set and change a port's settings.
- ReadFile() and WriteFile() for accessing a port.

See the documentation for your chosen language for details.

Under DOS, the process is very different. The remainder of this chapter describes DOS programming.

Initialization

Initializing the chip requires knowledge of the UART's register set. The first step is to set the baud rate divisor. You do this by first setting the DLAB (Divisor Latch Access Bit) high. This bit is Bit 7 at Base Address +3. In C code, the call would be:

```
outportb(BASEADDR +3,0x80);
```

You then load the divisor into Base Address +0 (low byte) and Base Address +1 (high byte). The following equation defines the relationship between baud rate and divisor:

desired baud rate = (UART Clock Frequency) / (32 * divisor)

On the PCI-COM485/8 card, the UART clock frequency is 1.8432 MHz. Below is a table for the popular divisor frequencies.

Baud Rate	Divisor x1	Divisor x8	Max Diff. Cable Length *
921600	N/A	1	250 ft
460800	N/A	2	500 ft
230400	N/A	4	800 ft
153600	N/A	6	1300 ft
115200	1	8	2200 ft
57600	2	16	4000 ft
38400	3	24	4000 ft
28800	4	32	4000 ft
19200	6	48	4000 ft
14400	8	64	4000 ft
9600	12	96	4000 ft
4800	24	192	4000 ft
2400	48	384	4000 ft
1200	96	768	4000 ft

*These are theoretical maximums based on typical conditions and good quality cables based on the EIA 485 and EIA 422 standard for balanced differential drivers.

Table 5-1: Baud Rate Divisor Values

In C, the code to set the chip to 9600 baud is:

```
outportb(BASEADDR +3, 0X80);// enters baud-divisor setup mode
outportb(BASEADDR, 0x0C);
outportb(BASEADDR +1,0);
```

The second initializing step is to set the Line Control Register at Base Address +3. This register defines word length, stop bits, parity, and the DLAB.

Bits 0 and 1 control word length and allow word lengths from 5 to 8 bits. Bit settings are extracted by subtracting 5 from the desired word length.

Bit 2 determines the number of stop bits. There can be either one or two stop bits. If Bit 2 is set to 0, there will be one stop bit. If Bit 2 is set to 1, there will be two stop bits.

Bits 3 through 6 control parity and break enable. They are not commonly used for communications and should be set to zeroes.

Bit 7 is the DLAB discussed earlier. It must be set to zero after the divisor is loaded or else there will be no communications.

The C command to set the UART for an 8-bit word, no parity, and one stop bit is:

```
outportb(BASEADDR +3, 0x03)
```

The final initialization step is to flush the receiver buffers. You do this with two reads from the receiver buffer at Base Address +0. When done, the UART is ready to use.

Reception

Reception can be handled in two ways: polling and interrupt-driven. When polling, reception is accomplished by constantly reading the Line Status Register at Base Address +5. Bit 0 of this register is set high whenever data are ready to be read from the chip. A simple polling loop must continuously check this bit and read in data as it becomes available. The following code fragment implements a polling loop and uses a value of 13, (ASCII Carriage Return) as an end-of-transmission marker:

```
do
{
 while (!(inportb(BASEADDR +5) & 1)); /*Wait until data ready*/
   data[i++]= inportb(BASEADDR);
}
 while (data[i]!=13);  /*Reads the line until null character rec'd*/
```

Interrupt-driven communications should be used whenever possible and is required for high data rates. Writing an interrupt-driven receiver is not much more complex than writing a polled receiver but care should be taken when installing or removing your interrupt handler to avoid writing the wrong interrupt, disabling the wrong interrupt, or turning interrupts off for too long a period.

The handler would first read the Interrupt Identification Register at Base Address +2. If the interrupt is for Received Data Available, the handler then reads the data. If no interrupt is pending, control exits the routine. A sample handler, written in C, is as follows:

```
readback = inportb(BASEADDR +2);
if (readback & 4)        /*Readback will be set to 4 if data are available*/
  data[i++]=inportb(BASEADDR);
outportb(0x20,0x20);          /*Write EOI to 8259 Interrupt Controller*/
return;
```

Transmission

RS485 transmission is simple to implement. The AUTO feature of the PCI-COM485/8 card automatically enables the transmitter when data is ready to send so no software enabling is required. The following software example is for non-AUTO operation.

To transmit a string of data, the transmitter must first check Bit 5 of the Line Status Register at Base Address +5. That bit is the transmitter-holding-register-empty flag. If it is high, the transmitter has sent the data. The process of checking the bit until it goes high followed by a write is repeated until no data remains.

The following C code fragment demonstrates this process:

```
outportb(BASEADDR +4, inportb(BASEADDR +4)|0x02);
  /*Set RTS bit without altering states of other bits*/
while(data[i]); /*While there is data to send*/
{
  while(!(inportb(BASEADDR +5)&0x20));    /*Wait until transmitter is empty*/
  outportb(BASEADDR,data[i]);
  i++;
}
outportb(BASEADDR +4, inportb(BASEADDR +4)&0xFD);
/*Reset RTS bit without altering states of other bits*/
```

Chapter 6: Connector Pin Assignments

Input/Output Connections

The PCI-COM485/8 Serial Communications card uses a 50-pin SCSI D-connector to interface to a spider cable. The spider cable has eight individual 9-pin connectors provided with it.

To ensure that there is minimum susceptibility to EMI and minimum radiation, it is important that the card mounting bracket be properly screwed into place and that there be a positive chassis ground. Also, proper EMI cabling techniques (cable connect to chassis ground at the aperture, shielded twisted-pair wiring, etc) be used for the input/output wiring.

PIN Number	RS-485 Signals	Pin Number	RS-485 Signals
Pin 1	GND Ground	Pin 26	GND Ground
Pin 2	Ain/out+	Pin 27	Ein/out+
Pin 3	Ain/out-	Pin 28	Ein/out-
Pin 4	GND thru 100 Ohm	Pin 29	GND thru 100 Ohm
Pin 5	Unused	Pin 30	Unused
Pin 6	Unused	Pin 31	Unused
Pin 7	GND Ground	Pin 32	GND Ground
Pin 8	Bin/out+	Pin 33	Fin/out+
Pin 9	Bin/out-	Pin 34	Fin/out-
Pin 10	GND thru 100 Ohm	Pin 35	GND thru 100 Ohm
Pin 11	Unused	Pin 36	Unused
Pin 12	Unused	Pin 37	Unused
Pin 13	GND Ground	Pin 38	GND Ground
Pin 14	Cin/out+	Pin 39	Gin/out+
Pin 15	Cin/out-	Pin 40	Gin/out-
Pin 16	GND thru 100 Ohm	Pin 41	GND thru 100 Ohm
Pin 17	Unused	Pin 42	Unused
Pin 18	Unused	Pin 43	Unused
Pin 19	GND Ground	Pin 44	GND Ground
Pin 20	Din/out+	Pin 45	Hin/out+
Pin 21	Din/out-	Pin 46	Hin/out-
Pin 22	GND thru 100 Ohm	Pin 47	GND thru 100 Ohm
Pin 23	Unused	Pin 48	Unused
Pin 24	Unused	Pin 49	Unused
Pin 25	GND Ground	Pin 50	GND Ground

Table 6-1: Connector Pin Assignments

Signal	Connector Pin
Unused	Pin 1
Ain/out+	Pin 2
Ain/out-	Pin 3
Unused	Pin 4
100 Ohm to Ground	Pin 5
Unused	Pin 6
Unused	Pin 7
Unused	Pin 8
Unused	Pin 9

Table 6-2: Data Cable Wiring

Appendix A: Application Considerations

Introduction

Working with RS422 and RS485 devices is not much different from working with standard RS232 serial devices and these two standards overcome deficiencies in the RS232 standard. First, the cable length between two RS232 devices must be short; less than 50 feet at 9600 baud. Second, many RS232 errors are the result of noise induced on the cables. The RS422 standard permits cable lengths up to 5000 feet and, because it operates in the differential mode, it is more immune to induced noise.

Connections between two RS422 devices (with CTS ignored) should be as follows:

Device #1		**Device #2**	
Signal	Pin No.	Signal	Pin No.
Gnd	7	Gnd	7
TX^+	24	RX^+	12
TX^-	25	RX^-	13
RX^+	12	TX^+	24
RX^-	13	TX^-	25

Table A-1: Connections Between Two RS422 Devices

A third deficiency of RS232 is that more than two devices cannot share the same cable. This is also true for RS422 but RS485 offers all the benefits of RS422 plus allows up to 32 devices to share the same twisted pairs. An exception to the foregoing is that multiple RS422 devices can share a single cable if only one will talk and the others will all receive.

Balanced Differential Signals

The reason that RS422 and RS485 devices can drive longer lines with more noise immunity than RS232 devices is that a balanced differential drive method is used. In a balanced differential system, the voltage produced by the driver appears across a pair of wires. A balanced line driver will produce a differential voltage from ±2 to ±6 volts across its output terminals. A balanced line driver can also have an input "enable" signal that connects the driver to its output terminals. If the "enable signal is OFF, the driver is disconnected from the transmission line. This disconnected or disabled condition is usually referred to as the "tristate" condition and represents a high impedance. RS485 drivers must have this control capability. RS422 drivers may have this control but it is not always required.

A balanced differential line receiver senses the voltage state of the transmission line across the two signal input lines. If the differential input voltage is greater than +200 mV, the receiver will provide a specific logic state on its output. If the differential voltage input is less than -200 mV, the receiver will provide the opposite logic state on its output. A maximum operating voltage range is from +6V to -6V allows for voltage attenuation that can occur on long transmission cables.

A maximum common mode voltage rating of ±7V provides good noise immunity from voltages induced on the twisted pair lines. The signal ground line connection is necessary in order to keep the common mode voltage within that range. The circuit may operate without the ground connection but may not be reliable.

Parameter	Conditions	Min.	Max.
Driver Output Voltage (unloaded)		4V	6V
		-4V	-6V
Driver Output Voltage (loaded)	LD and LDGND	2V	
	jumpers in	-2V	
Driver Output Resistance			50Ω
Driver Output Short-Circuit Current			±150 mA
Driver Output Rise Time			10% unit interval
Receiver Sensitivity			±200 mV
Receiver Common Mode Voltage Range			±7V
Receiver Input Resistance			4KΩ

Table A-2: RS422 Specification Summary

To prevent signal reflections in the cable and to improve noise rejection in both the RS422 and RS485 mode, the receiver end of the cable should be terminated with a resistance equal to the characteristic impedance of the cable. (An exception to this is the case where the line is driven by an RS422 driver that is never "tristated" or disconnected from the line. In this case, the driver provides a low internal impedance that terminates the line at that end.)

Note

You do not have to add a terminator resistor to your cables when you use the PCI-COM485/8 card. Termination resistors for the RX+ and RX- lines are provided on the card and are placed in the circuit when you install the LD and LDGND jumpers. (See the Option Selection section of this manual.)

RS485 Data Transmission

The RS485 Standard allows a balanced transmission line to be shared in a party-line mode. As many as 32 driver/receiver pairs can share a two-wire party line network. Many characteristics of the drivers and receivers are the same as in the RS422 Standard. One difference is that the common mode voltage limit is extended and is +12V to -7V. Since any driver can be disconnected (or tristated) from the line, it must withstand this common mode voltage range while in the tristate condition.

The following illustration shows a typical multidrop or party line network. Note that the transmission line is terminated on both ends of the line but not at drop points in the middle of the line.

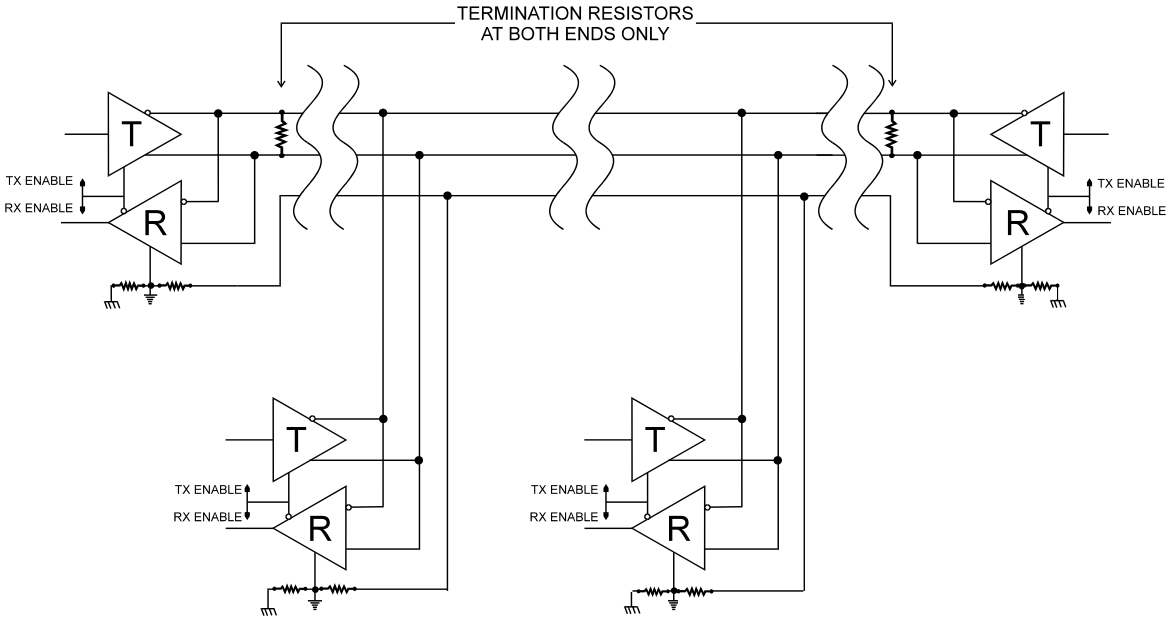

Figure A-1: Typical RS485 Two-Wire Multidrop Network

RS485 Four-Wire Multidrop Network

An RS485 network can also be connected in a four-wire mode. In a four-wire network it's necessary that one node be a master node and all others be slaves. The network is connected so that the master communicates to all slaves and all slaves communicate only with the master. This has advantages in equipment that uses mixed protocol communications. Since the slave nodes never listen to another slave's response to the master, a slave node cannot reply incorrectly.

Customer Comments

If you experience any problems with this manual or just want to give us some feedback, please email us at: ***manuals@accesioproducts.com.***. Please detail any errors you find and include your mailing address so that we can send you any manual updates.

ACCES I/O PRODUCTS, INC.
10623 Roselle Street, San Diego CA 92121
Tel. (858)550-9559 FAX (858)550-7322
www.accesioproducts.com